

100 ナノ量子電気伝導計算むけの並列量子波束ダイナミクス ソルバー開発

井町宏人

鳥取大学

1. はじめに

フレキシブルデバイスなどに豊富な応用をもつ有機半導体材料のシミュレーションには、100 ナノメートルスケールの非理想構造計算が必要となる。本稿では量子ダイナミクス法による有機高分子系の量子電気伝導計算と、その並列実装の性能評価を示す。

量子電気伝導の支配方程式は**時間依存 Schrodinger 型方程式**

$$i\dot{\psi} = H(t)\psi \quad (1)$$

である。ここで $\psi(t) \in \mathbf{C}^m$ は電子波束の状態ベクトルであり、 $H(t) \in \mathbf{R}^{m \times m}$ はハミルトニアン行列と呼ばれる実対称疎行列である。 $H(t)$ はその時刻の原子構造から決定され、行列次元 m はほぼ系の原子数 M に比例する。時間依存 Schrodinger 型方程式を解いて電子波束の時間発展を求め、種々の物理量を得る手法を量子ダイナミクス法と呼ぶ。100 ナノメートルスケール系を扱う場合原子数は $M = 10^8$ のオーダーに及び、また量子ダイナミクス法による伝導性能の算出のためには 10^4 時間ステップ以上の長時間のシミュレーションが必要となるため、ソルバーには高い並列性能が要求される。

本稿では、原子構造を決定する量子 MD の計算およびハミルトニアン行列 $H(t)$ の生成には、独自開発の超並列電子状態計算コード ELSESES [1, 2, 3, 4] を用いる。ELSESES は原子数 M に比例する計算量で実行可能な量子 MD アルゴリズムを実装しており、「京」コンピューター全 (82, 994) ノードまでの高い強スケーリング性能の実績がある[5]。その際の MD 計算速度は「京」コンピューター 1 計算ノードあたり 500 原子において、1 MD 時間ステップあたり 10 秒程度であった。一般に、原子の動きを追跡する MD の時間ステップ幅に比べて、電子波束計算の時間ステップ幅はより小さく取らなければならない。本稿ではその比を 1/10 とし、量子 MD の計算時間と同程度、すなわち 1 (電子)時間ステップあたり 1 秒を量子ダイナミクス法の計算速度の目標値とする。

なお、本来は時間依存 Schrodinger 型方程式は重なり行列と呼ばれる正定値実対称行列 S を用いて

$$iS(t)\dot{\psi} = H(t)\psi$$

と書かれるが、本稿では簡単のため $S = I$ とし扱う。 S はもともと単位行列に近く、正定値対称であるという性質があるため、方程式の性質は大きく変化しない。また、本稿の内容は 3.3 節を除き $S \neq I$ の場合にも直接適用可能である。

2. 時間発展計算の連立一次方程式への帰着

本稿では(1)式の時間発展計算を行うために、**Crank-Nicolson 法**による離散化を行う [6]。すなわち、時刻 t において、時間刻み幅 Δt に対して

$$\psi(t + \Delta t) - \psi(t) = \frac{\Delta t}{2} \{H(t + \Delta t)\psi(t + \Delta t) + H(t)\psi(t)\}$$

とする。1 時間ステップで行うべき計算を整理すると、

$$\psi(t + \Delta t) = \left\{I + \frac{i\Delta t}{2}H(t + \Delta t)\right\}^{-1} \left\{I - \frac{i\Delta t}{2}H(t)\right\}\psi(t)$$

である。この離散化は、 H が時間に依存しない場合に限り、時間 2 次精度となり、またノルム $\|\psi(t)\|_2$ が時間に対して保存されるという優れた性質を持つ。ノルム保存性は物理的な意味のある要請であり、それが離散化後も保たれていることは本質的に重要である。元の方程式 (1) がノルム保存性を持つことは H の実対称性から従う性質であり、

$$\frac{d}{dt} \|\psi(t)\|_2^2 = \frac{d\psi^\dagger(t)}{dt} \psi(t) + \psi^\dagger(t) \frac{d\psi(t)}{dt} = i\psi^\dagger(t)H(t)\psi(t) - i\psi^\dagger(t)H(t)\psi(t) = 0$$

で確認できる。

本稿の設定では H は時間依存のためこれらの性質は満たされないが、実際には H の時間変化は緩やかなため Crank-Nicolson 法は有力な解法である。この離散化によって解法は陰解法となり、計算量の大半は、連立一次方程式

$$Ax = b \quad (2)$$

を解く部分が占める。ここで、

$$A = I + \frac{i\Delta t}{2}H(t + \Delta t), x = \psi(t + \Delta t), b = \left\{I - \frac{i\Delta t}{2}H(t)\right\}\psi(t)$$

である。

係数行列 A は複素であるが、 H は実であるという構造を持つため一つの複素行列として扱うのは無駄が大きい。そのため、以降は複素連立一次方程式 (2) を実部と虚部に分け、 $n = 2m$ 次元の実連立一次方程式として扱う。その際係数行列 A は、複素ベクトル x, b を $x \rightarrow x + iy, b \rightarrow b + ic$ と書き直して、

$$\left\{I + \frac{i\Delta t}{2}H(t + \Delta t)\right\}(x + iy) = b + ic$$

$$\left(x - \frac{\Delta t}{2}H(t + \Delta t)y\right) + i\left(y + \frac{\Delta t}{2}H(t + \Delta t)x\right) = b + ic$$

より

$$A = \begin{pmatrix} I & -\frac{\Delta t}{2}H(t + \Delta t) \\ \frac{\Delta t}{2}H(t + \Delta t) & I \end{pmatrix}$$

と書ける。こうすることで、一つの複素行列として扱った場合に比べて、行列ベクトル積の計算量を約半分にする。前述の通り $H(t)$ は疎行列であり、時間的に緩やかに変化する。以下では、これらの性質を利用して、連立一次方程式を高速に解く方法を検討する。

3. 連立一次方程式の前処理付き反復解法と並列実装

3. 1. 前処理付き GMRES 法

本稿では非対称疎行列 A を係数行列とする連立一次方程式の求解に、広く採用されている手法である前処理付き GMRES 法を用いる。左前処理行列 P を用いた GMRES アルゴリズムは以下の通りである [8]。内積を (\cdot, \cdot) で表記する。

Algorithm (Left-Preconditioned GMRES)

1. 初期解 x_0 を選び, $r_0 := P(b - Ax_0), \beta := \|r_0\|_2, v_1 := r_0 / \beta$ とする。
2. For $j = 1, \dots, m$ do
3. $w := PA v_j$
4. For $i = 1, \dots, j$ do
5. $h_{i,j} := (w, v_i)$
6. $w := w - h_{i,j} v_i$
7. End For
8. $h_{j+1,j} := \|w\|_2, v_{j+1} := w / h_{j+1,j}$
9. End For
10. $V_m := [v_1, \dots, v_m], \bar{H}_m := \{h_{i,j}\}_{1 \leq i \leq j+1, 1 \leq j \leq m}$
11. $y_m := \operatorname{argmin}_y \|\beta e_1 - \bar{H}_m y\|_2, x_m := x_0 + V_m y_m$ とし, x_m を解とする。

ここで, $e_1 \in \mathbf{R}^{m+1}$ は第 1 要素のみが 1 で他が 0 であるベクトルである。上記アルゴリズムでは反復回数 m は固定回数のように書かれているが, 実際には m は固定ではなく, 残差ノルム等で収束と判定した時点で反復を停止させることが可能である。

連立一次方程式の数値解法の性能は, しばしば前処理の選択に大きく依存する。扱う問題・計算機環境に合わせた前処理の設計が重要である。前処理の追加による総計算コストの増減は次のようにまとめられる;

1. 前処理行列 P の計算のためのコスト増加,
2. 前処理行列 P との疎行列ベクトル積を追加することによるコスト増加,
3. 反復回数 m の減少によるコスト減少。

本稿の問題設定では係数行列 A が時間的に緩やかに変化する複数の連立一次方程式を解くので, 再利用可能な前処理の場合, 時間ステップ数が大きくなるほど 1. のコスト増加分の重みは小さくなる。一方, 2. と 3. のコスト変化分は時間ステップ数の増大の影響を受けづらい。2. について, 疎行列ベクトル積の計算量は疎行列の非ゼロ要素数に比例する。また, 3. について, 反復解法の反復回数は係数行列の条件数が小さいほど少なくなる。以上から, 非ゼロ要素数が小さく, 前処理後の係数行列 PA の条件数なるべく小さくなるような前処理行列 P をつける必要がある。

3. 2. 近似逆行列前処理

並列化に適した前処理手法として, 近似逆行列 (AINV, Approximate Inverse) 前処理が挙げられる [8]。これはまず, 前処理後の係数行列の条件数をなるべく小さくするという要請を, 目的関数

$$F(P) = \|I - AP\|_F^2$$

の最小化として定式化する。これを近似的に解く方法は様々であるが, ここではフロベニウスノルムを列ごとに分解することで

$$F(P) = \sum_{j=1}^n \|e_j - Ap_j\|_2^2$$

と変形し、列のインデックス j ごとに連立一次方程式 $Ap_j = e_j$ を近似的に解くことで、前処理行列 $P = [p_1, \dots, p_n]$ とする（列ベクトル形式の AINV 前処理）。ただし、この連立一次方程式 1 本を解くのに $O(n)$ の計算量をかけると全体として $O(n^2)$ の計算量となり、元の問題の計算量を上回ってしまう。このため、それぞれの連立一次方程式は行列だけでなくベクトルについても疎性を利用して解く必要がある。ベクトルの疎性を保つため、反復の初期解は疎とし、反復ごとに絶対値の小さい要素を 0 に置き換える操作（numerical dropping）を行う。

列ベクトル形式の AINV 前処理行列は、列のインデックス j ごとに独立に計算可能なため、並列計算に向いている。現実的な設定では $n \gg$ (総並列スレッド数) であるため、理想的な並列化が可能である。また、一度計算した AINV 前処理行列は、数値的に近い係数行列に対して再利用が可能である。これは、現在の「係数行列 A が時間的に緩やかに変化する複数の連立一次方程式を解く」という問題設定において非常に有効に働く。 A が前処理行列を作成した時点での値からずれるに従って前処理行列の質が相対的に悪化し、反復回数が増加した際は、適当な時点で前処理行列を作り直すことが考えられるが、本稿では実装していない。

3. 3. Chebyshev 多項式前処理

次に、Chebyshev 多項式による前処理について解説する。これは多項式 s と元の係数行列 A を用いて $P = s(A)$ とする、すなわち元の問題 $Ax = b$ の代わりに

$$s(A)Ax = s(A)b$$

を解くものである。ここで、 $s(A)$ と A は交換することに注意する。前処理後の係数行列 $s(A)A$ の条件数を最小にする k 次多項式 s は次で定まる。

$$\text{Find } s \in P_k \text{ that minimizes } \max_{\lambda \in \sigma(A)} |1 - \lambda s(\lambda)|$$

ここで P_k は k 次までの多項式全体、 $\sigma(A)$ は A の固有値全体の集合である。しかし、これは A の固有値全体を求める部分で元の問題よりも難しい問題を解いているため、不適切である。代わりに、 λ の動く範囲を $\sigma(A)$ を包含する領域 E に置き換えて、

$$\text{Find } s \in P_k \text{ that minimizes } \max_{\lambda \in E} |1 - \lambda s(\lambda)|$$

を解くことにする。 E は $\sigma(A)$ になるべく近いことが望ましい。 E を複素平面上の楕円に取ったとき、この最小化問題の解は Chebyshev 多項式によって与えられ、行列ベクトル積 $s(A)x$ は k 回までの A との疎行列ベクトル積を通じて効率的に計算可能である。詳細は文献 [8] に譲る。

現在扱っている係数行列 A のブロック構造に着目すると、 $B := -\frac{\Delta t}{2}H(t + \Delta t)$ として、

$$\det(\lambda I - A) = \det \begin{pmatrix} (\lambda - 1)I & -B \\ B & (\lambda - 1)I \end{pmatrix} = \det\{(\lambda - 1)^2 I + B^2\}$$

であるから、 A の固有値は B の固有値 $\lambda_j (j = 1, \dots, m) \in \mathbf{R}$ を用いて $1 \pm i\lambda_j$ と書ける。 B の固有値の存在範囲も系が定まれば精度よく推定可能なので（小さな部分系について固有値問題を実際に解いてみてもよい）、 A の固有値は 1 から虚軸方向に、ある推定可能な範囲だけ移動したうちに入っている。これを利用して、 $\sigma(A)$ を精度よく近似する楕円領域 E を決めることができる（図 1）。

Chebyshev 多項式による前処理は、前処理行列の事前計算が必要なく、係数行列 A とその固有値分布を包含する楕円領域 E のみの情報から適用できる。このため、係数行列が緩やかに変化する場合、固有値分布も緩やかに変化すると考えられるので、再利用が可能である。 k 次

多項式の前処理を利用する場合，疎行列ベクトル積の回数で計算量を見積もると，「2. 前処理行列 P との疎行列ベクトル積を追加することによるコスト」が $k+1$ 倍になるので，3. の反復回数 m が $k+1$ 分の 1 以下になる場合に意味を持つ。

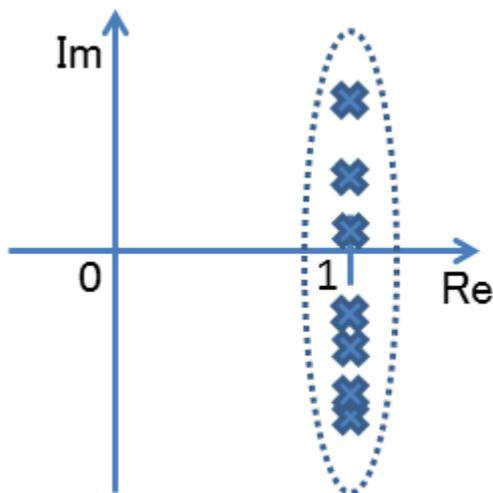


図 1: 複素平面上の，係数行列 A の固有値 (×印) と，それを包含する楕円 (点線)。

3. 3. FX10 における並列実装

以下，並列実装の詳細について説明する。並列化は FX10 上で行った。FX10 は 16 コアの SPARC64 IXfx CPU 1 つを 1 ノードとし，ノード間を Tofu インターコネクトで接続したスーパーコンピュータである。本稿ではプログラミング言語に Fortran を用い，ノード間の並列化を MPI で，ノード内の並列化を OpenMP で行った。以降，ノード数 (=MPI プロセス数) を P_N ，コア数 (= OpenMP スレッド数) を $P_T (= 16)$ で表記する。

並列実装の単純化のため，主反復 (GMRES 法) においては，疎行列形式には Coordinate list (COO) 形式を採用した。一方，AINV 前処理行列の生成のための連立一次方程式を解く部分では，疎行列-疎ベクトル積を効率的に行うため，疎行列は疎な列ベクトルを並べた形式で表現した。

AINV 前処理においては n 本の連立一次方程式を $P_N \times P_T$ 個のスレッドに均等に振り分け，各々は非並列で解いた。前述の通り，各々が独立であるため通信が発生せず，現実的には $n \gg P_N \times P_T$ が満たされるため，理想的な並列化が可能である。

主反復においては COO 形式の疎行列 A とベクトル x の疎行列ベクトル積

1. $y(:) = 0$
2. For $k = 1, \dots, A.nnz$ do
3. $i = A.is(k)$
4. $j = A.js(k)$
5. $a = A.vals(k)$
6. $y(i) += a \cdot x(j)$
7. End For

に対し、インデックスの範囲 $k = 1, \dots, A.nnz$ を $P_N \times P_T$ 個に均等に分割する単純な並列化を行った。ここで $A.nnz$ は A の非ゼロ要素数である。実装の簡単のため、それ以外の部分は非並列である。

また、入力は大規模な行列データとなるので、全行列データを単一のノードが読み込み、それを各ノードに分配するという実装だと、読み込みがボトルネックになる可能性がある。そこで、代表的な疎行列データ形式である MatrixMarket 形式 [9] をベースに、複数ファイルに分割可能にした行列データ形式とその並列入出力ルーチンを実装した。また、高速化のため、データはバイナリ形式とした。FX10 のエンディアンはビッグエンディアンであるため、リトルエンディアンの環境で作成した行列データはそのままでは非互換になってしまうが、以下のエンディアンを強制する open 命令、

```
open(file=path, unit=i, access="stream", form="unformatted", convert="little_endian")
```

は FX10 の Fortran コンパイラでも有効だったため、これを利用した。

4. 数値実験

高分子系の量子電気伝導計算の実問題から連続した行列を生成し、それに対して上記の並列連立一次方程式ソルバーを利用したシミュレーションを適用した。系は高分子材料の一種である poly-(phenylene-ethynylene) (PPE) の一本鎖、12,000 原子である (図 2)。行列次元は $m = 35994$ ($n = 71988$) 次元、非ゼロ要素数は時間ステップによって多少変動するが、約 1,880,000 個である。

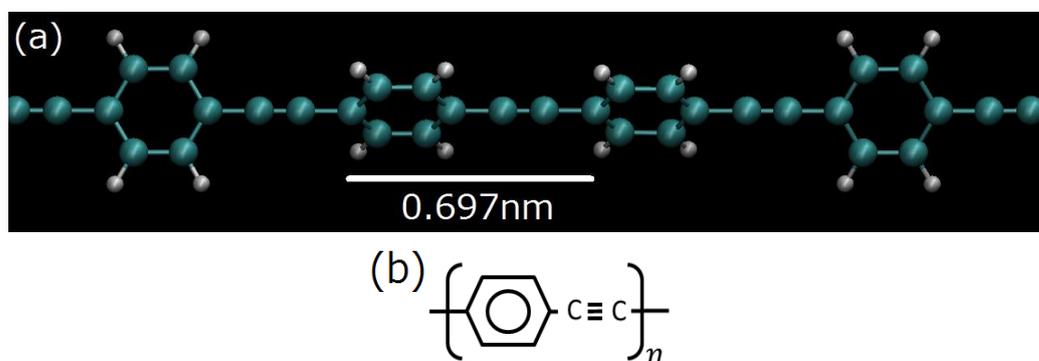


図 2: (a) poly-(phenylene-ethynylene) (PPE) の構造 (一部を拡大), (b) 同化学式。 n は任意回数の繰り返しを表し、文中で使用している 12,000 原子系では $n = 1,000$ であり、(a) はそのうちの $n = 4$ 個分を抜き出して拡大したものである。ベンゼン環のつなぎ目では回転が可能となっている。

PPE に対する量子 MD ・ 量子波束ダイナミクスの例として [5], 量子電気伝導計算の例として [7] が挙げられる。

手法の各パラメーターは以下の通りである。全体のシミュレーション時間は 4000 a.u. (≈ 100 fsec), 行列の (すなわち MD の) 時間刻み幅は 40 a.u. (≈ 1 fsec), 電子波束の時間刻み幅は 4 a.u. (≈ 0.1 fsec) とした。すなわち, 10 時間ステップごとに係数行列 A が時間発

展し、その間は A は一定である。AINV 前処理の numerical dropping のスレッシュホールド（絶対値がこの値を下回った疎ベクトルの要素は反復ごとに 0 にする）は 1.0×10^{-2} ，連立一次方程式の反復解法の反復回数は 2 回とした。主反復（GMRES 法）の収束判定基準は残差ノルム 1.0×10^{-12} ，最大反復回数 m_{\max} は 100 回とした。FX10 の 3 から 48 ノードまでを利用して評価を行った。

なお、この系に対しては、Chebyshev 多項式前処理が前処理なしの解法の計算速度を上回った例がなかったため、以降の結果を省略し、AINV 前処理のみ扱う。

まず、48 ノード計算における、AINV 前処理行列の計算時間と、1 時間ステップあたりの平均反復回数、平均計算時間を表 1 にまとめた。AINV 前処理の導入によって反復回数が低下し、長時間シミュレーションで重要な 1 ステップあたりの平均計算時間は、前処理なしで 3.537 秒だったのが 0.673 秒と短縮され、目標としていた 1 ステップあたり 1 秒以内の時間発展計算が実現されていることが分かる。AINV 前処理行列の計算時間は 11.696 秒で、1 ステップあたりの平均計算時間の短縮によって得られる全体の計算時間の減少量に比べて十分に小さい。

表 1: AINV 前処理行列の作成時間 (Time(AINV)) および、AINV 前処理なし/ありによる、1 時間ステップあたりの反復回数 (AvgIter(step))・計算時間 (AvgTime(step)) の変化。Time(AINV) + AvgTime(step) × ステップ数が総計算時間となる。

	Time(AINV) [sec]	AvgIter(step)	AvgTime(step) [sec]
前処理なし	-	147.777	3.537
AINV 前処理あり	11.696	47.867	0.673

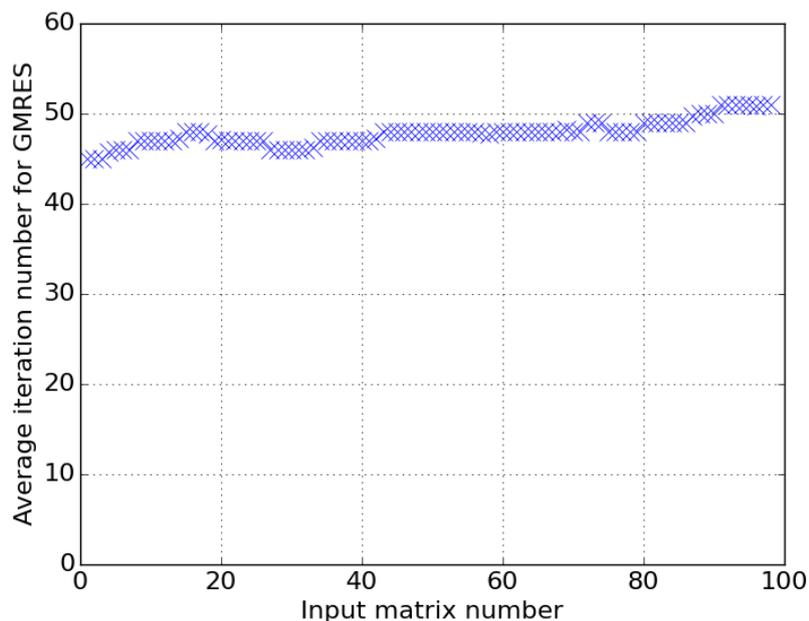


図 3: 行列の時間発展に伴う、AINV 前処理ありでの GMRES 法の反復回数の変化。横軸は時間発展による行列の番号、縦軸はその時点の行列に対する 10 時間ステップ（本文参照）の時間発展計算で平均した、GMRES 法の反復回数。

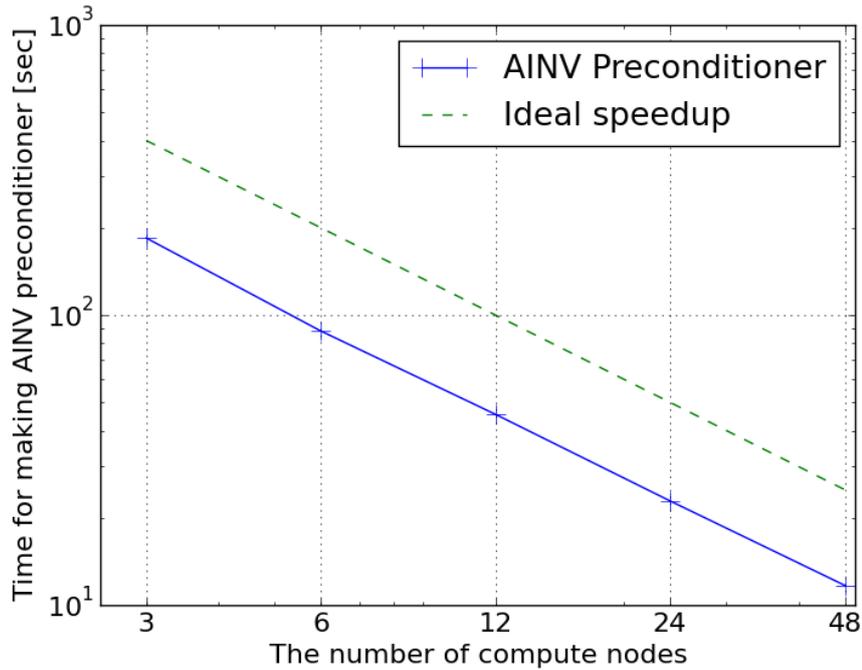


図 4: 計算ノード数(横軸)に対する AINV 前処理行列の計算時間(縦軸)の両対数プロット。実線が実際の計算時間で、破線は理想的な速度向上の傾きを示す。

以下, AINV 前処理について詳細に見ていく。行列の時間発展に伴って, GMRES 法の反復回数が増える様子を図 3 に示す。AINV 前処理行列を作成した時点では反復回数が 45 回ほどだったのが, 100 番目 (100 fs 経過後) の行列に対しては, 50 回強と悪化している。ただし, 悪化速度は非常に緩やかであり, AINV 前処理行列の再利用がうまく働いていることが分かる。

最後に, AINV 前処理行列計算の強スケーリング性能を図 4 に示す。非常に良好なスケーリングが実現されている。最大のノード数 $P_N = 48$ においても, 連立一次方程式の総数 $n = 71988$ 本は総スレッド数 $P_N \times P_T = 768$ に比べて十分大きいので, 理想的な並列性能が発揮されている。

5. まとめ

大規模な非理想系に対する量子電気伝導計算を目的とした並列量子ダイナミクスソルバーを開発した。時間依存 Schrodinger 方程式を Crank-Nicolson 法で離散化することで, 係数行列が緩やかに変化する複数の連立一次方程式に帰着する。これらを連続して解くための, 並列前処理付き反復解法を実装し, FX10 上で性能を確認した。

今後の展望としては以下が挙げられる。

本稿での性能評価は比較的小規模な行列を対象としたものに留まったので, 100 ナノメートルスケールの系から生成される行列 ($m = 10^8$ 次元以上) に対する性能評価が必要である。並列疎行列ベクトル積を始めとする並列反復解法の各部分の最適化は多くの研究があり, ライブラリ実装も数多く存在する。本稿での実装は簡便なものであり, これら既存のライブラリを適宜活用することで性能の向上が見込める。また, 係数行列が緩やかに変化する複数の連立一次方程式に対して, 前処理を(方程式ごとに毎回前処理を作り直すのに比べて低いコストで) 微修

正しつつ再利用する手法が研究されており[10], より長時間のシミュレーションにおいては, それら前処理の修正手法と前処理の作り直しを適切に組み合わせることで, 前処理の効果を高められると考えられる。

本稿では線形な Schrodinger 方程式とその Crank-Nicolson 法による数値解法を扱ったが, 近い問題設定として, 電子波束の自己集中を扱う非線形 Schrodinger 方程式に対しても種々の数値解法が提案されている。中でも, 元の微分方程式のノルム保存性, エネルギー保存性などの性質を保ったまま離散化を行う構造保存数値解法は有力な解法クラスのひとつである[11, 12]。それらに現れる連立一次方程式に対しても, 本稿で開発した手法は有効であると考えられ, 興味深い適用先である。

また, 本稿で開発したライブラリはインターネット上で公開予定である。その際, 前述の MatrixMarket 形式をベースにしたバイナリデータ形式の並列入出力ルーチン等, 同時に開発したユーティリティー類も合わせて公開する。

謝 辞

本研究は, 東京大学情報基盤センター「スーパーコンピューター若手・女性利用者推薦」平成 27 年度 (前期・後期) 課題によって行われた。また, 同課題の支援制度を用い, 片桐孝洋先生 (現名古屋大) との有用な議論を行った。

参 考 文 献

- [1] ELSESES (= Extra Large-Scale Electronic Structure calculation); <http://www.elses.jp/>
- [2] T. Hoshi, S. Yamamoto, T. Fujiwara, T. Sogabe, and S.-L. Zhang, “An order- N electronic structure theory with generalized eigenvalue equations and its application to a ten-million-atom system,” *J. Phys. Condens. Matter* 24, 165502, 5pp. (2012).
- [3] T. Hoshi, Y. Akiyama, T. Tanaka, and T. Ohno, “Ten-million-atom electronic structure calculations on the K computer with a massively parallel order- N theory,” *J. Phys. Soc. Jpn.* 82, 023710, 4pp. (2013).
- [4] 星健夫, 「超大規模量子物質シミュレーターELSESES の FX10 および「京」での超並列計算」, 東京大学情報基盤センター「スーパーコンピューティングニュース」, Vol. 15 No. 4, pp. 32-41, (2013 年 7 月号).
- [5] T. Hoshi, H. Imachi, K. Kumahata, M. Terai, K. Miyamoto, K. Minami, and F. Shoji, “Extremely scalable algorithm for 10^8 -atom quantum material simulation on the full system of the K computer,” to appear in *Proceeding of ScalA16: Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, held in conjunction with SC16, Nov. 13, 2016, Salt Lake City, UT, USA.
- [6] T. Tada, T. Yamamoto, and S. Watanabe, “Molecular orbital concept on spin-flip transport in molecular junctions; wave-packet scattering approach and Green’s function method,” *Theor. Chem. Acc.* 130: pp. 775-488 (2011).
- [7] J. Terao, A. Wadahama, A. Matono, T. Tada, S. Watanabe, S. Seki, T. Fujihara, and Y. Tsuji, “Design principle for increasing charge mobility of π -conjugated polymers using regularly localized molecular orbitals,” *Nat. Commun.* 4, 1691 (2013).

- [8] Y. Saad, “Iterative Methods for Sparse Linear Systems,” 2nd ed., SIAM, Philadelphia, PA, USA (2003).
- [9] <http://math.nist.gov/MatrixMarket/>
- [10] J. D. Tebbens and M. Tũma, “Efficient Preconditioning of Sequences of Nonsymmetric Linear Systems,” SIAM J. Sci. Comput., 29(5), pp. 1918–1941 (2007).
- [11] Y. Miyatake and J. C. Butcher, “A characterization of energy-preserving methods and the construction of parallel integrators for Hamiltonian systems,” SIAM J. Numer. Anal., 54 pp. 1993–2013 (2016).
- [12] 井町宏人, 宮武勇登, 星健夫「エネルギー保存数値解法の大規模物質計算への展開」, 日本応用数理学会 研究部会連合発表会, 神戸, 2016年3月.